High-level synthesis and arithmetic optimizations

Yohann Uguen, Florent de Dinechin, Steven Derrien

High-level synthesis and arithmetic optimizations

Yohann Uguen and Florent de Dinechin and Steven Derrien

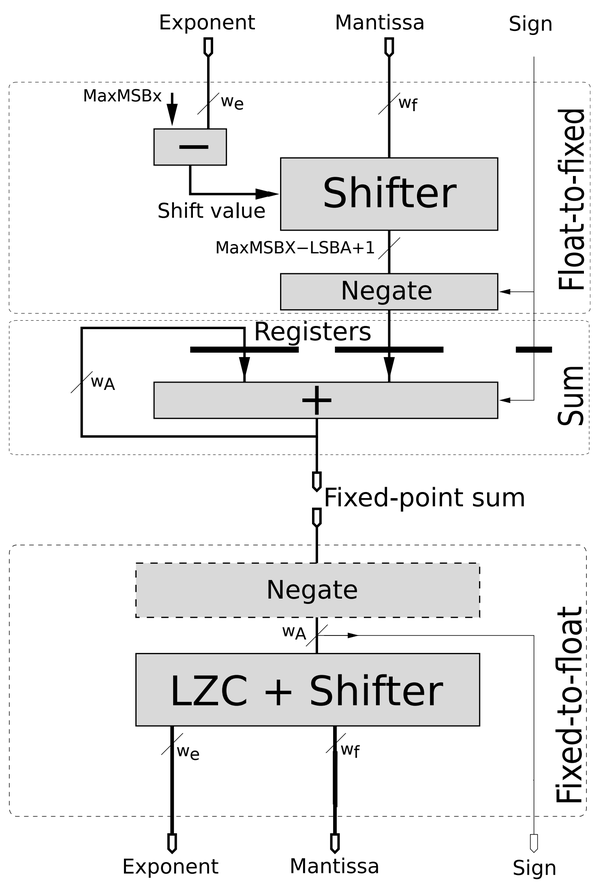
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON — citi lab — Inria informatiques mathématiques — Université de Rennes 1

Context

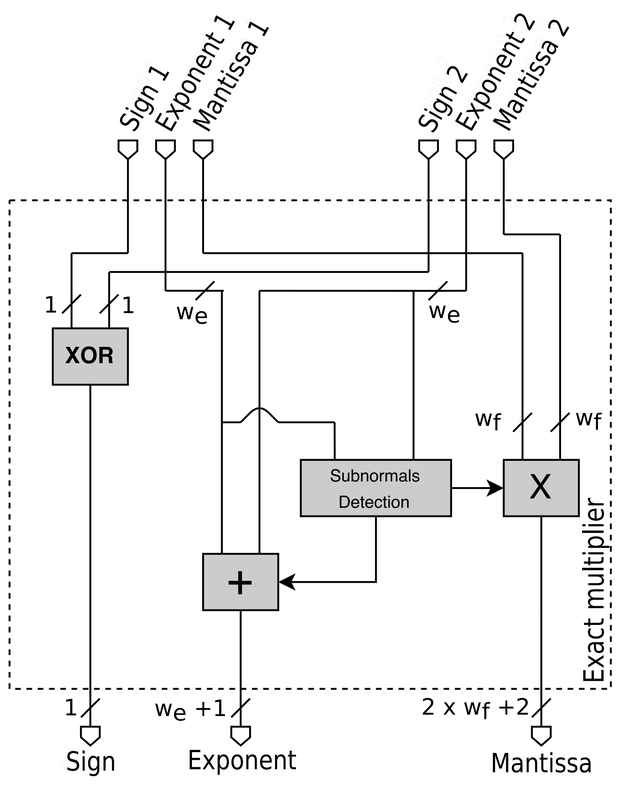
- Computing with real numbers
- Design a hardware accelerator
- Targeting FPGAs
- Trade off between
  1. performance
  2. accuracy
  3. resource usage
  4. ease of use

Accumulator

Based on Kulisch and Snyder accumulator [3]



Exact Multiplier



References

[1] de Dinechin, Florent et al.: *An FPGA-specific Approach to Floating-Point Accumulation and Sum-of-Products*, FPT 2008

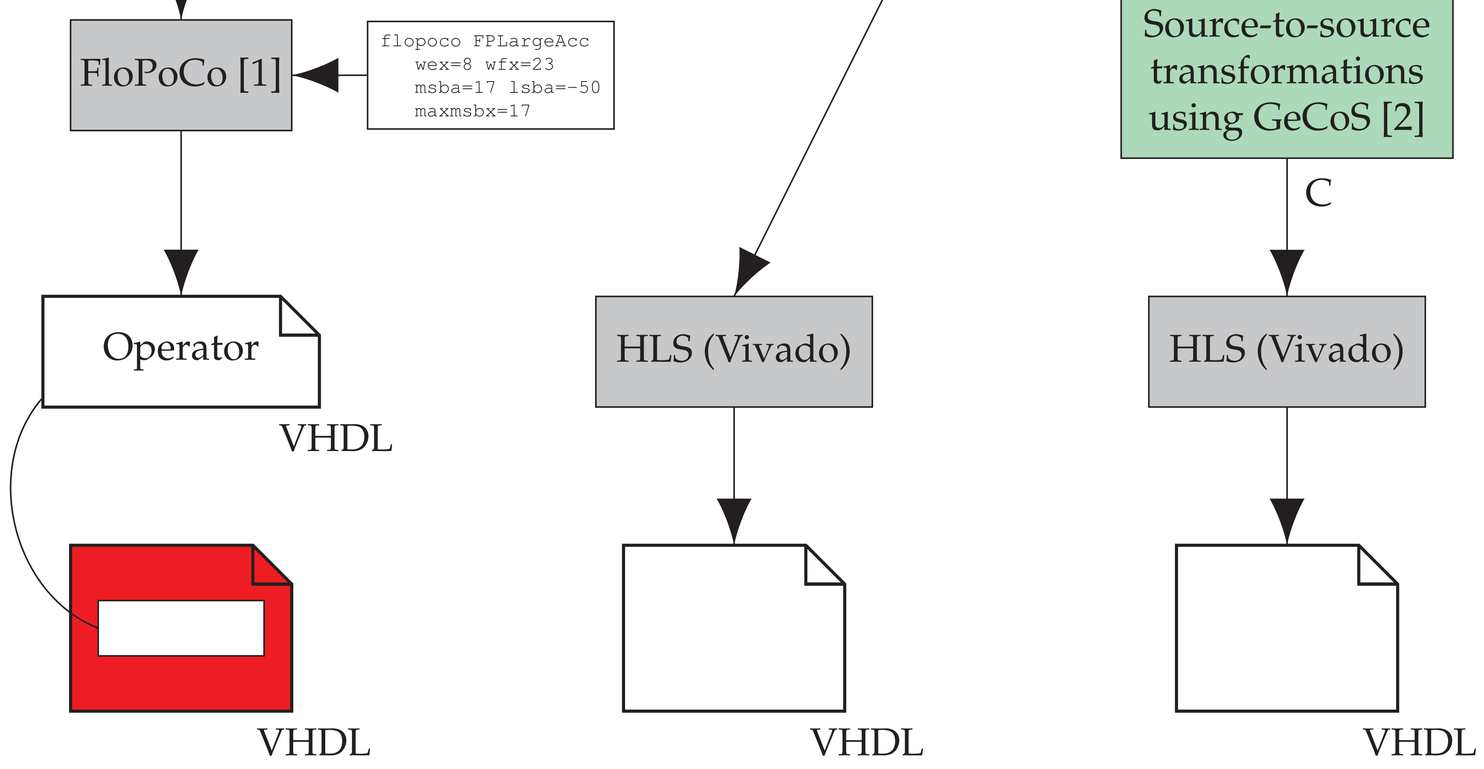
[2] Floc'h, Antoine et al.: *GeCoS: A framework for prototyping custom hardware design flows*, SCAM 2013

[3] Kulisch, Ulrich and Snyder, Van: *The Exact Dot Product As Basic Tool for Long Interval Arithmetic*, Computing 2011

Merging application-specific arithmetic and HLS

$$\sum_i x_i \times y_i$$

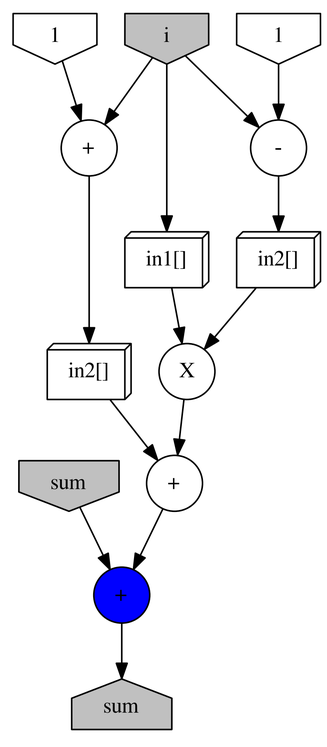
```
float sumOfProduct(float in1[N], float in2[N])
  float sum = 0;
  #pragma FPacc VAR=sum MaxAcc=100000 epsilon=1E-15
  for (int i=0; i<N; i++)
    sum+=in2[i+1];
    sum+=in1[i]*in2[i-1];
  return sum;
```

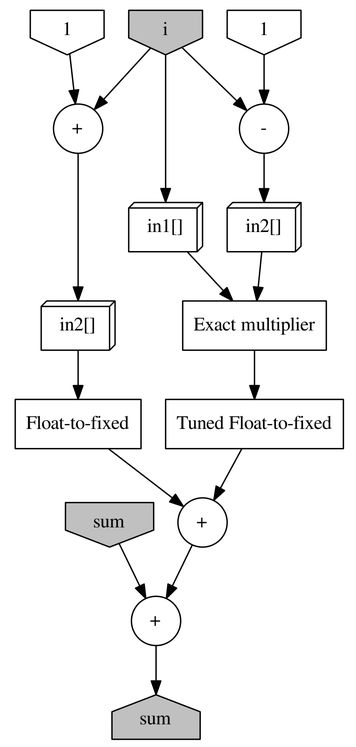


FloPoCo	HLS (Vivado)	Source-to-source + HLS
⊕ Low resource	⊖ Moderate resource	⊕ Low resource
⊕ Low latency	⊖ Moderate latency	⊕ Low latency
⊕ Accuracy control	⊖ No accuracy control	⊕ Accuracy control
⊖ Difficult to use	⊕ Ease of use	⊕ Ease of use

Source-to-source transformations using GeCoS

switch *Node* **do**
  case +
    Launch recursively on incoming nodes
  end
  case ×
    Replace with exact multiplier combined with tuned Float-to-fixed operator
  end
  case *Accumulation variable*
    Ignore
  end
  otherwise
    Insert Float-to-fixed node
  end
endsw





Results

Input values in $[0, 1]$
100K accumulations

	FloPoCo's Operator	Naive Code	Transformed Code
Accumulator width	67	24	67
LUTs	693	313	868
DSPs	2	5	2
Latency	100K	1000K	100K
Accuracy	24 bits	17 bits	24 bits